AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°2; pp 372-390
Submitted Mar. 05, 2017; Revised May 15, 2017; Accepted Jun. 10, 2017

Prediction Method of Network Security Situation Based on GA-LSSVM Time Series Analysis

*Huan Wang, **Jian Gu, ***Jianping Zhao*, ****Dan Liu, *****Xin Sui, ******Xiaoqiang Di, *******Bo Li

*School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
**School of Opto-electronic Engineering, Changchun University of Science and Technology, Changchun 130022, China
***School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China (471745553@qq.com)
****School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
*****School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
******School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
*******School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China

Abstract

To more accurately understand the development trend of network security situation and to solve the prediction problem in network security situation awareness, this paper proposes a prediction model and an optimization method of network security situation based on GA-LSSVM time series analysis. The model adopts the original sequence accumulation method to reduce the interference of the irregular fluctuations of the original sequence and constructs the mixed kernel function based on the combination of RBF and Poly which takes both the learning and generalization ability of the model into account. The genetic algorithm is used to optimize the parameters of the LSSVM model. Through characteristic chromosome coding of the model parameters, the search space is established to obtain the optimal solution through fitness evaluation.

The simulation results show that the model can effectively predict the network security situation with an accuracy of about 13% higher than that of HHGA-RBFNN and PSO-SVM.

Key words

LSSVM, Trend prediction, Parameter optimization.

1. Introduction

Network security situation awareness (NSSA) is a kind of network situation awareness (CSA), which refers to the acquisition, understanding, display and forecast of future development trend of network security elements. By considering various security factors, the network security situation is dynamically reflected on the whole to forecast and forewarn the development trend of the future security situation. It has become a research popularity in the field of network security, attracting enough attention of relevant personnel. Network security situation awareness includes three parts: security elements extraction, quantification and forecasting, among which situation prediction is the key and difficult point of the situation awareness technology. Situation prediction is mainly to predict the network security situation in a future period of time based on the current network security situation value so that the corresponding response measures can be taken before the attack on the network. After the situation assessment, the network security situation can be reflected by a series of situational values, which are equivalent to an array of time series, so situational prediction is equivalent to time series prediction.

In recent years, many foreign and domestic scholars have conducted in-depth studies on the network security situation prediction and achieved some results. Huang Tongqing [1] proposes a network security situation prediction model HMM-NSSP based on the hidden Markov model from the perspective of combining theory with practice and presents the method for network security situation prediction. To more accurately evaluate and forecast the network security situation, Guo Chunxiao [2] explores the factors affecting the global convergence performance of the algorithm on the basis of the study of Quantum Particle Swarm Optimization (QPSO), forming an improved QPSO algorithm based on evolutionary strategy. Through the research on Trusted Network Connection Framework (TNC) and Network Situation Perception System (CSA), WU Kun [3] proposes the set-pair situation assessment and forecast method (SPSAF) of network security aiming at the certainty and uncertainty of multi-data source in trusted network security. Tang Chenghua et al. [4] proposes a network security situation prediction method based on the likelihood of BP, but the parameter training process of the method is relatively complicated with a low convergence speed. Huang Tongqing et al. [1] propose a network security situation prediction model HMM-

NSSP based on the hidden Markov model from the perspective of combining theory with practice and present the method for network security situation prediction. Wu Shuyue et al. [6] apply the hidden Markov model to the recognition of abnormal user behavior and their algorithm extracts the features from the user's shell command sequence. Liu Yuling et al. [7] proposes a network security situation prediction method based on time-space dimension analysis, which extracts the evaluation factors from three aspects of the attacker, the protector and the network environment and analyzes the security situation factor sets and the effect of their mutual influence on the network security situation in a spatial dimension, thus obtaining the network security situation.

Based on the above research and aimed at the prediction problem of network security situation awareness, this paper proposes a network security situation prediction method based on GA-LSSVM time series analysis according to the non-linear characteristics of network security situation values. The method first constructs the training set and establishes LSSVM prediction model; and then the genetic algorithm with characteristic chromosome is employed to optimize the parameters of the LSSVM model; and finally, the optimized parameters are used to re-adjust the prediction model for the situation prediction.

2. The LSSVM-based Prediction Model

2.1 Model Definition

The training samples sets of the network security situation are assumed to be (x_1, y_1), ..., (x_i, y_i), ..., (x_n, y_n), where $x_i \in R^m$, $y_i \in R$, i=1, ..., n, x_i is the input vector and y_i is the output vector. The principle of LSSVM-based prediction is to convert the nonlinear function relation of the input space into the linear function relation in the high dimensional feature space through the nonlinear function $\phi(x)$. The regression function adopted for this model is shown in Formula (1).

$$y(x) = w^T \phi(x) + b \tag{1}$$

Where $\phi: R^m \rightarrow H$, ϕ is the feature map, H is the feature space, w is the weight vector in the space H, and $b \in R$ is the offset quantity. According to the principle of structural risk minimization, Formula (1) is transformed into a quadratic optimization problem:

$$\min J(w,e) = (w^T \cdot w)/2 + \gamma(\sum_{i=1}^{n} e_k^2)/2 \tag{2}$$

The constraint condition:

$$y_k = w^T \phi(x_k) + b + e_k \quad \text{k} = 1, \cdots, n \tag{3}$$

Where γ is the adjustable regularization parameters, and e_k is the error variable.

According to the duality theory, the Lagrange equation corresponding to (2) is constructed:

$$L(w,b,e,\alpha) = \frac{1}{2} w^T \cdot w + \frac{1}{2}\gamma(\sum_{i=1}^{n} e_k^2) - \sum_{i=1}^{n} \alpha_i [w^T \phi(x_i) + b + e_i - y_i] \tag{4}$$

Where α_i is the Lagrange multiplier. In accordance with the KKT optimization condition, Formula (4) is used to obtain the partial derivatives of w, b, e, α, respectively, and let it be 0 to obtain the system of linear equations (5):

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(x_1,x_1)+\frac{1}{\gamma} & \cdots & K(x_1,x_n) \\ M & M & O & M \\ 1 & K(x_n,x_1) & \cdots & K(x_n,x_n)+\frac{1}{\gamma} \end{bmatrix} \bullet \begin{bmatrix} b \\ \alpha_1 \\ M \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ M \\ y_n \end{bmatrix} \tag{5}$$

According to the Mercer condition, the kernel function (6) is defined:

$$K(x_K, x_l) = \phi^T(x_k)\phi(x_l) \tag{6}$$

The inner product operation in the high dimensional space is converted into the functional calculation in the input space. Solve the value of α and b in (3), and the LSSVM regression model is

$$y_t = \sum_{t,j=1}^{n} \alpha_j K(x_K, x_i) + b \tag{7}$$

2.2 Kernel Function Selection

The common kernel functions are RBF and Poly as shown in equations (8) and (9),

respectively. The RBF kernel function is a local kernel function, which is better at describing local than global characteristics of nonlinear problems, while the Poly kernel function is a global kernel function, which is better at describing global than local characteristics.

$$K_{RBF}(x, x_i) = \exp\left[-\frac{|x - x_i|^2}{2\sigma^2}\right] \tag{8}$$

$$K_{Poly}(x, x_i) = (x^T, x_i + 1)^d \tag{9}$$

To take into account both the learning and generalization ability of the model and to enhance prediction accuracy, a kernel function combining RBF and Poly is proposed in this paper as shown in equation (10).

$$K_{Mix}(x, x_i) = \lambda K_{RBF}(x, x_i) + (1 - \lambda) K_{Poly}(x, x_i) \tag{10}$$

2.3 Data Processing

The network security situation value is characterized with non-linearity and irregularity, and SVM is sensitive to data between 0~1 with high training speed. Therefore, this paper first accumulates the original network security situation values $\{x^{(t)}(1), x^{(t)}(2), \ldots, x^{(t)}(n)\}$ at the moment t to obtain the new network security situation values $\{x^{(t^*)}(1), x^{(t^*)}(2), \ldots, x^{(t^*)}(n)\}$, and the accumulation process is shown in equation (11).

$$x^{(t^*)}(k) = \sum_{i=1}^{k} x^{(t)}(i) \quad k = 1,2,\cdots,n \tag{11}$$

Then, the security situation values are normalized, as shown in equation (12)

$$x^{(t^*)}(i)^* = \frac{x^{(t^*)}(i) - x^{(t^*)}(\min)}{x^{(t^*)}(\max) - x^{(t^*)}(\min)} \tag{12}$$

Where $x^{(t^*)}(i)$ is the accumulated value, $x^{(t^*)}(\max 1)$ and $x^{(t^*)}(\min)$ are the maximum and minimum accumulated value, respectively. In restoring data, we first conduct denormalization according to the formula (13) and then carry out continuous subtraction according to the formula

(14) to acquire the final result.

$$x^{(t^*)}(i) = x^{(t^*)}(i)^* \times (x^{(t^*)}(\max) - x^{(t^*)}(\min)) + x^{(t^*)}(\min) \tag{13}$$

$$x^{(t)}(i+1) = x^{(t^*)}(i+1) - x^{(t^*)}(i) \qquad i = n, n+1, n+2, \cdots \tag{14}$$

3. GA-based LSSVM Joint Parameter Optimization

There are four parameters of the hybrid kernel LSSVM to be determined, i.e. the penalty parameter γ, the polynomial order d, the kernel width σ and the adjustable parameters λ. γ is used to balance the training error and complexity of LSSVM, exerting a great influence on the promotion ability of LSSVM. d determines the complexity of calculation, and the higher value of d will greatly increase the computational complexity of the Poly kernel function. σ reflects the corresponding width of the inner product towards the input data, whose value affects the distribution of sample data in high dimensional feature space. λ controls the comprehensive performance of the mixed kernel function by adjusting the proportion of Poly and RBF kernel function. The four parameters interactively and commonly influence the prediction accuracy of the mixed kernel LSSVM, so the four parameters should be simultaneously optimized during the parameter selection. Genetic algorithm is an adaptive optimization search method based on the simulation of Darwin's natural selection and biological system inheritance, which guides the search direction by the fitness function and the probability transformation rule. The genetic algorithm can be applied to the feature selection and parameter optimization of the hybrid kernel LSSVM to realize the global optimization of the parameter combination (γ, d, σ, λ) of the mixed kernel LSSVM and improve the prediction accuracy.

Based on the genetic algorithm, the proposed genetic algorithm with characteristic chromosome is fundamentally different from the non-characteristic chromosome genetic algorithm in that: it integrates the asymptotic performance of support vector machine when applied to the feature selection and parameter optimization of support vector machine and generates characteristic chromosome through characteristic chromosome manipulation. The fitness evaluation is conducted on the newly generated characteristic chromosomes and progeny chromosomes to select the chromosomes with high fitness as the next generation of population, thus continuing the genetic manipulation and characteristic chromosome manipulation of the next generation.

3.1 Chromosome Coding

The variable values and the feature subset selection *f* of the support vector machine parameters (*γ, d, σ, λ*) are converted into binary encoding. The chromosomal coding of the genetic algorithm with characteristic chromosome consists of two parts: the parameters encoding and the feature subset selection coding, as shown in Fig.1.

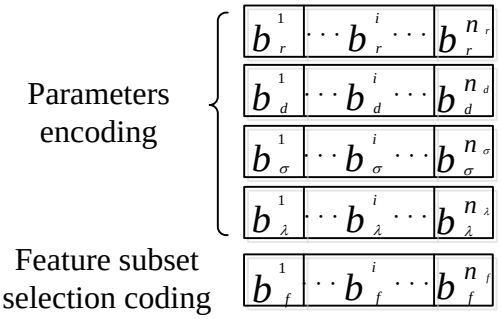

Fig.1. Chromosome Coding

In the coding of the parameter pairs (*γ, d, σ, λ*), $b_\gamma^1 \sim b_\gamma^{n_\gamma}$, $b_d^1 \sim b_d^{n_d}$, $b_\sigma^1 \sim b_\sigma^{n_\sigma}$, $b_\lambda^1 \sim b_\lambda^{n_\lambda}$ are the binary codes of the parameters of *γ, d, σ, λ,* respectively; n_γ, n_d, n_σ, n_λ are the binary bits of the parameters *γ, d, σ, λ*; n_σ and n_γ are selected into the coding of the feature subset selection *f* according to the calculation accuracy, and 1 means the feature's being selected and 0 indicates not being selected; $b_f^1 \sim b_f^{n_f}$ is the binary code of the feature subset selection *f*, where n_f is its binary bit and equal to the characteristic number of the data set.

3.2 Selection and Replication

In the genetic process, the individual with greater fitness has greater probability of being selected. During the operation of replication, a number of chromosomes with the largest fitness values are selected as the male parent to be directly inherited to the next generation.

The expected value of the individual is solved by the expectation value method:

$$E_i = \frac{f_i}{f_{sum} / N} \tag{15}$$

Where f_i is the fitness of the individual i, f_{sum} is the total fitness of the population, and N is the population size.

The individual expectation value determines whether the individual in the population enters the next generation for optimization, and the number of the replicated individuals i is E_i. The initialized population changes from p to p^*after selection and replication.

3.3 Crossover and Variation

The crossover operation is independently conducted on the parent through N in the current population $p(t)$ to generate a new population $p^*(t)$, and the crossover operation is conducted on the parameters (γ, d, σ, λ) and the feature subset f, respectively. The crossover follows the crossover rule of binary code: one-point crossover operation, which means to randomly set a cross point in the individual series and to exchange the partial structure of the two individuals x_1 and x_2 before and after the cross point during the implementation of crossover to generate two new individuals. The specific process is shown in equation (16).

$$\begin{cases} y_1 = \alpha x_2 + (1-\alpha)x_1 \\ y_2 = \alpha x_1 + (1-\alpha)x_2 \end{cases} \tag{16}$$

where $\alpha \in [0,1]$ is a random number. The crossover rate and the mutation rate adopt the adaptive selection, and the crossover probability $P_c$and the mutation probability P_c are automatically changed with the fitness according to (17) and (18).

$$p_c = \begin{cases} \dfrac{\tau_1(fit_{\max} - fit^*)}{fit_{\max} - \bar{fit}} & fit^* \geq \bar{fit} \\ \tau_3 & fit^* < \bar{fit} \end{cases} \tag{17}$$

$$p_m = \begin{cases} \dfrac{\tau_2(fit_{\max} - fit\ )}{fit_{\max} - \bar{fit}} & fit\ \geq \bar{fit} \\ \tau_4 & fit\ < \bar{fit} \end{cases} \tag{18}$$

where $fit_{\max}$ is the maximum fitness of the current population, $\bar{fit}$ is the average fitness of the generation of population, fit^* is the one with larger fitness in the parents to be crossed, and $fit$is the fitness of variation individuals. And the value range of τ_1, τ_2, τ_3, τ_4 is set to be (0,1), and $\tau_1=\tau_3=1$, $\tau_2=\tau_4=1/2$, in this paper.

3.4 Fitness Function

The fitness function consists of support vector machine classification accuracy, selected feature number and characteristic cost. The high fitness depends on high classification accuracy, small feature number and low characteristic cost, so the fitness function is defined as (19) to avoid the denominator's approaching zero.

$$fit = w_A A + w_F \left(P + F_i \sum_{i=1}^{n_f} C_i \right)^{-1} \tag{19}$$

where A is the classification accuracy; W_A is the weight of the classification accuracy, which is set by the users generally in the range of 0.75 ~ 1; P is a constant set to avoid the denominator's approaching zero, generally between 1~10; C_i is the characteristic cost, and the relevant data set of UCI has different characteristic costs, which can be set to the same value between 1~8 if there is no characteristic cost information; F_i is the characteristic value, which equals to 1 when the i^{th} feature is selected and equals to 0 when it is not; W_F is the feature weight and W_F =1-W_A.

3.5 Generation of Characteristic Chromosome Manipulation

The progressive performance of the Gaussian kernel SVM proposed by Keerthi et al. [8] states that the hyper-parametric space can be established with its parameters being logγ and logσ^2. In the asymptotic region of the space, there exists a generalization error contour line, separating the hyper-parametric space into two regions: an over-fitting or under-fitting region, and a fitting region. The fitting region is most likely to possess the hyper-parameter set with the best generalization error. When $\sigma^2 \to \infty$ along the contour line, the best penalty parameter $\tilde{\gamma}$ satisfies the formula (20).

$$\log \sigma^2 = \log \gamma - \log \tilde{\gamma} \tag{20}$$

In the fitting region of the hyper-parameter space (logγ, logσ^2), the appropriate value of $\tilde{\gamma}$ can be selected to obtain the optimal generalization error line, i.e. the optimal line of the highest accuracy, thus finding the optimal parameter. The difficulty is how to select the appropriate value of $\tilde{\gamma}$.

This paper proposes that the search function of genetic algorithm can be applied to select the r chromosomes with the highest fitness value in each generation for the appropriate choice of the

value of $\tilde{\gamma}$. From the hyper-parameter space to the genetic algorithm environment, the equation (21) is transformed into:

$$\tilde{\gamma} = \gamma / \sigma^2 \tag{21}$$

For the sake of convenient expression, the equation (22) can be obtained by replacing $\tilde{\gamma}$ with K and $1/\sigma^2$ with μ.

$$K = \gamma * \mu \tag{22}$$

Equation (22) is an important formula for the application of the asymptotic performance of support vector machines into genetic algorithms.

In each of the selected r chromosomes, the parameter pair (γ, μ) is taken for coding and converted into the corresponding variable values, and the K values are calculated by Eq. (22), respectively. The search range of the variable value of the parameter γ is discretized into n values, and the d values of the parameter μ are calculated by the formula (21) to generate $r \times n$ new parameter pairs. And then they are converted into binary codes to connect with the corresponding feature selection codes of the original selected chromosomes, totally generating $r \times n$ new chromosomes:

$$(\gamma_{11}, \mu_{11}, f_1),...,(\gamma_{1n}, \mu_{1n}, f_1),...,(\gamma_{21}, \mu_{21}, f_2),...,(\gamma_{2n}, \mu_{2n}, f_2),...,(\gamma_{r1}, \mu_{r1}, f_r),...,(\gamma_{rn}, \mu_{rn}, f_r)$$

The newly generated chromosomes select the appropriate value (i.e. the K value), which contains the feature of asymptotic performance of the support vector machine, so the generated chromosomes are called characteristic chromosomes. The current and evolutionary generations of the characteristic chromosomes contain the characteristic chromosomes including the optimal parameter γ and parameter μ. The procedure for generating a characteristic chromosome is as follows:

Step1: Select the parent to generate the characteristic chromosome. The number of the generated characteristic chromosomes is initialized to be f_c and that of discrete values in the search range of the variable value of parameter γ is n, so the number of chromosomes with the highest fitness value to be selected should be $r = f_c/n$. The chromosomes are arranged in a decreasing order in accordance with the fitness value of the current generation after crossover and variation, and the

r chromosomes with the highest fitness value are selected to serve as the male parent of the characteristic chromosomes. The parameter pairs (γ, μ) of the male parent chromosomes are extracted for coding and converted into the corresponding variable values.

Step2: Calculate the K value of each parent

$$K_{\mathrm{i}} = \gamma_{\mathrm{i}} * \mu_{\mathrm{i}} \quad i = 1,2,...,r. \tag{23}$$

Step3: Discrete the search range of the variable value of parameter γ into n values, so the n values of the parameter γ_i are

$$\gamma_{i1}, \gamma_{i2}, \cdots, \gamma_{in} \quad i = 1,2,...,r. \tag{24}$$

Step4: Calculate the value of the corresponding parameter μ by K and γ with the formula

$$\mu_{ij} = K_i / \gamma_{ij} \quad i = 1,2,...r,\ j = 1,2,...,n \tag{25}$$

Step5: Generate the characteristic chromosomes. The variable values of the $r \times n$ generated parameter pairs are converted into binary codes and then connected to the corresponding feature selection codes in the original selected chromosomes to generate $r \times n$ characteristic chromosomes.

In the process of generating characteristic chromosomes, the asymptotic performance of the support vector machine is incorporated into the genetic algorithm to generate the characteristic chromosome population by calculating the $\tilde{\gamma}$ value of the chromosomes with high fitness value in the offspring population M (t). The generated characteristic chromosome operator $T_f: M(t) \rightarrow F(t)$ enhances the search efficiency of the genetic algorithm and improves the classification accuracy of the support vector machine.

3.6 Joint Optimization Algorithm

Algorithm 1. GA-based LSSVM Parameter Optimization Algorithm

Input: the network security situation data set $\{x(1), x(2), \cdots, x(n)\}$ Output: the penalty parameter γ, the polynomial order d, the kernel width σ, and the adjustable parameter λ 1. Establish the training sample set and the test sample set based on the input data set

2. Select the feature subset and carry out normalization with $x^{(t^*)}(i)^* = \dfrac{x^{(t^*)}(i) - x^{(t^*)}(\min)}{x^{(t^*)}(\max) - x^{(t^*)}(\min)}$

3. Build SVM classifier

4. Calculate the classification accuracy

5. Conduct chromosome coding on $\gamma, d, \sigma, \lambda$ according to the method in 3.1

6. Initialize the population

7. Calculate and evaluate the fitness according to the formula $fit = w_A A + w_F \left(P + F_i \sum_{i=1}^{n_f} C_i \right)^{-1}$

8. **while** ($fit > \varepsilon$)**do**

9. Calculate $fit = w_A A + w_F \left(P + F_i \sum_{i=1}^{n_f} C_i \right)^{-1}$

10. Carry out crossover and variation according to the formula $\begin{cases} y_1 = \alpha x_2 + (1-\alpha) x_1 \\ y_2 = \alpha x_1 + (1-\alpha) x_2 \end{cases}$

11. Generate new characteristic chromosomes according to the algorithm in 3.4

12. Carry out selection and replication based on the calculation results of $E_i = \dfrac{f_i}{f_{sum} / N}$

13. Generate the new population p^*

14. **end while**

15. Decode the chromosomes and reduce the parameters $\gamma, d, \sigma, \lambda$

16. Output the penalty parameter γ, the polynomial order d, the kernel width σ, and the adjustable parameter λ

4. GA-LSSVM-based Prediction

The basic idea of GA-LSSVM prediction is that: the training set is first constructed; then the algorithm 1 is used for parameter optimization based on the training set; and finally the prediction model is reconstructed with the optimized parameters to forecast the situation. The process of the GA-LSSVM-based network security situation prediction is:

Step1: Calculate the network security situation value by the evaluation model.

Step2: Carry out data normalization according to formulas (11) and (12).

Step3: Reconstruct the network security situation data, generate the security situation sample data set with the time series method and divide the security situation sample set into training sample set and test sample set. The training sample set is trained by the support vector machine to obtain the initial prediction model, and the test sample set is used to detect the prediction accuracy of the initial prediction model. This paper adopts the form of open-set test, which means ensuring no intersection and complete independence between the two during the division of the training sample set and the test sample set.

Step4: Input the constructed training sample set into the GA-LSSVM model and search the

optimal training parameters (γ, d, σ, λ) of the support vector machine by the genetic algorithm with characteristic chromosome.

Step5: Determine whether the prediction accuracy of the parameters reaches the standard. Execute Step6 if the standard is reached; otherwise, execute Step4.

Step6: Establish support vector machine for training to generate the final prediction model.

Step7: Input the test sample into the final prediction model to obtain the safety situation value.

Step8: Denormalize the output results according to the formulas (13) and (14) for reduction of the results.

5. Experimental Simulation

5.1 Experimental Environment

To verify the rationality and correctness of the network security situation prediction model and algorithm proposed in this paper, the DARPA99 intrusion detection data set is employed to select the training and test data with the replay tool being Netpoke. The experimental environment adopts the network topology used in literature [9], and the host is configured with an 8-core CPU, 8GB memory and a Gigabit NIC. The development environment is Matlab 7.3, and the support vector machine software is LIBSVM.

The simulation network will operate for two weeks (February 12, 2016 to March 2, 2016). To maintain the generality, large sample and small sample data experiments are carried out, respectively. The large sample data contains a total of 480 sets of data with the sampling period being 0.5 hours, and the small sample data calculate the daily security situation to obtain 48 sets of data. The two experiments only differ in data with the process being completely consistent, thus further verifying the accuracy of the model. For the two sets of data, the former 50% are selected into the training set and the latter 50% are selected into the test set. The network security situation values are calculated with the method in literature [9].

The algorithm 1 is adopted to optimize the parameters of LSSVM (γ, d, σ, λ), among which the search range of γ is (0.01, 35000), the search range of σ is (0.0001, 32), d is an integer between [1, 10] and λ is a random number in [0, 1]. In the process of chromosome coding, the parameters $\gamma, d, \sigma, \lambda$ are orderly arranged, the length is $n_\gamma+n_d+n_\sigma+n_\lambda$ and the search space is $2^{n\gamma+nd+n\sigma+n\lambda}$. The crossover and variation probabilities are calculated with equations (17) and (18).

5.2 Evaluation Methods

To further demonstrate the superiority of the method, the absolute error, the average absolute

error, the root mean square error and the average relative error are separately employed for its evaluation. The calculation method of absolute error is shown in equation (26).

$$E = \left| s_i' - s_i \right| \tag{26}$$

The average absolute error is calculated by the equation (27)

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \frac{| s_i - s_i' |}{s_i} \tag{27}$$

The root mean square error is calculated by the equation (28)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (s_i - s_i')^2} \tag{28}$$

The average relative error is calculated by the equation (29)

$$ARVE = \sum_{i=1}^{N} [s_i - s_i']^2 \Big/ \sum_{i=1}^{N} [s_i - \bar{s}_i]^2 \tag{29}$$

Where s_i' is the predicted value, s_i is the actual value and $\bar{s}_i$ is the average of the actual values.

5.3 Result Analysis

The training set of the large sample experiment data is input into SVM for learning, and the genetic algorithm with characteristic chromosome is used to optimize the data. The curve of the fitness changing along with the increase of iterations is shown in Fig.2.

It can be seen from Fig. 1 that the first group of experiments achieves the optimal results when the iterations reach 60 times, obtaining the optimal LSSVM parameters: γ=100, d=7, σ=*0.01*, λ=0.3. These parameters are employed to reconstruct the LSSVM model for the situation forecast, and the predicted trend values are shown in Fig 3.

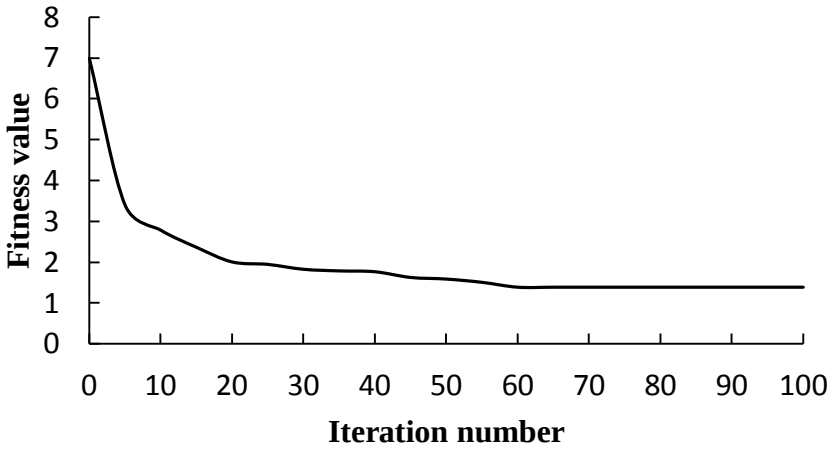

Fig.2. LSSVM Parameter Optimization Curve

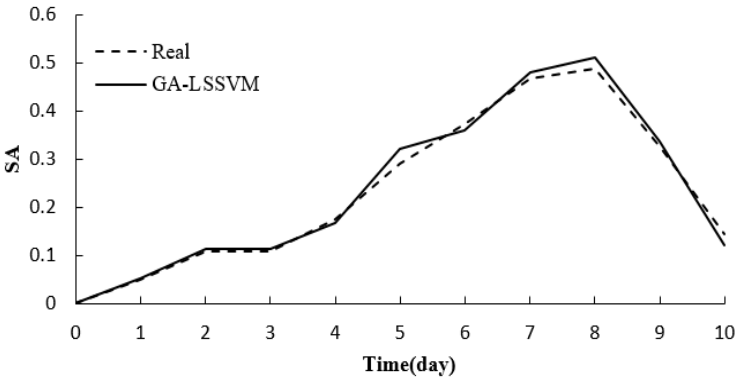

Fig.3. The Predicted Situation Values of the Overall Data Set in the First Set of Data on the Network Security Situation (28 points)

To verify the superiority of GA-LSSVM in prediction accuracy, it is compared with the existing network security situation prediction algorithms HHGA-RBFNN and PSO-SVM through the same experiments. The result is obtained as shown in Fig.4.

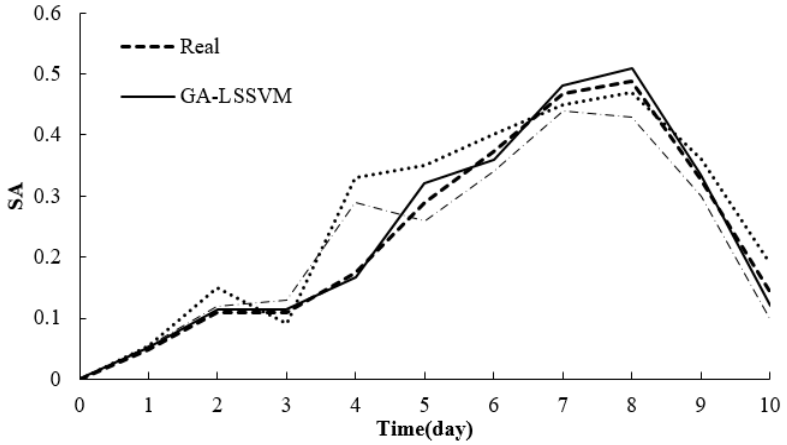

Fig.4. Comparison of Predicted Values for Different Algorithms

The experimental results of Fig.4 show that being closer to the actual security situation with smaller fluctuations, the GA-LSSVM prediction algorithm manifests obvious advantages compared with the existing network security situation prediction algorithms HHGA-RBFNN and PSO-SVM. To better evaluate the prediction accuracy, the absolute error index E, the average absolute error *MAPE*, the root means square error *RMSE* and the average relative error *ARVE* are used to evaluate its performance.

First, the absolute error index E is used to evaluate the predicted values of the four methods, obtaining the results shown in Fig.5. It can be seen from Fig.5 that the absolute error of GA-LSSVM is significantly less than that of the other two methods.

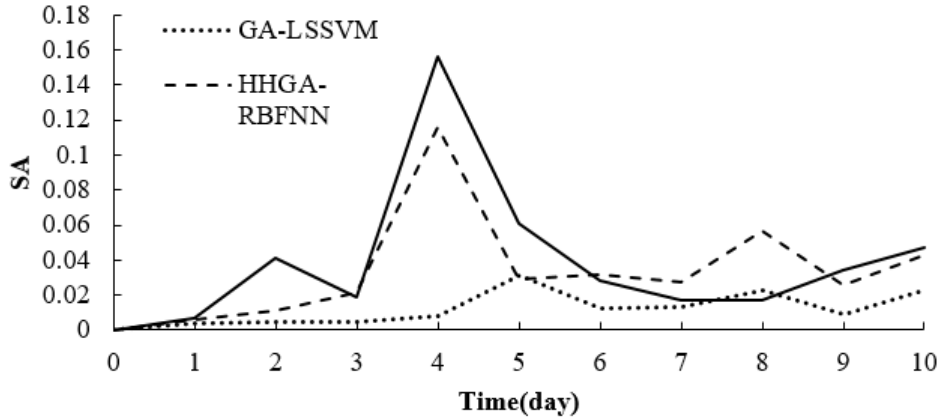

Fig.5. Absolute Error Indicators of the Three Models

Then, they are further evaluated with the average absolute error *MAPE*, the root means square error *RMSE* and the average relative error *ARVE*. Obviously, smaller error indicates higher prediction accuracy.

As can be seen from Table 1, the three error values of the prediction by the GA-LSSVM method are significantly smaller than those of the other two methods, which are closer to the actual security situation values.

Considering the universality of the model, the small sample data experiment is carried out to predict the daily situation. The former 24 sets of data are training set, and the latter 24 sets of data are test set. The errors are evaluated with *MAPE* and *RMSE* to obtain the experimental results as shown in Table 2.

The two groups of experiments with the large and small samples show that the GA-LSSVM method proposed in this paper possesses higher prediction accuracy and more obvious advantages than the other two algorithms.

Tab.1. Comparison of Relative Accuracy of the Three Methods

Algorithms	*MAPE*	*RMSE*	*ARVE*
PSO-SVM	0.062408967	0.015946787	0.011394778
HHGA-RBFNN	0.182692504	0.047351874	0.100469413
GA-LSSVM	0.238346123	0.059055059	0.156269324

Tab.2. Results of the Small Sample Data Experiment

No	$(\gamma, d, \sigma, \lambda)$	GA-LSSVM		HHGA-RB		PSO-SVM	
		RMSE	*MAPE*	*RMSE*	*MAPE*	*RMSE*	*MAPE*
1	（95,5,0.02,0.35）	0.02083	0.06946	0.04212	0.13179	0.05315	0.20323
2	（82,6,0.03,0.48）	0.02042	0.06636	0.04072	0.1429	0.05672	0.18308
3	（79,9,0.09,0.35）	0.02001	0.06326	0.03932	0.15401	0.06029	0.16293
4	（84,3,0.02,0.35）	0.01967	0.06016	0.03792	0.16512	0.06386	0.14278
5	（69,5,0.10,0.35）	0.01919	0.05706	0.03652	0.17623	0.06743	0.12263
6	（73,7,0.02,0.35）	0.01878	0.05396	0.03512	0.18734	0.07213	0.10248
7	（55,4,0.07,0.35）	0.01895	0.06396	0.03752	0.18878	0.07421	0.99237
8	（48,3,0.11,0.35）	0.01912	0.07396	0.03992	0.18934	0.07742	0.08567
9	（67,2,0.04,0.35）	0.01929	0.08396	0.04232	0.19034	0.08063	0.17215
10	（91,6,0.02,0.35）	0.01714	0.04156	0.02952	0.23178	0.08528	0.05988

Conclusion

Aimed at the prediction problem in network security situation awareness, this paper proposes a prediction model and an optimization method of network security situation based on GA-LSSVM time series analysis. The model adopts the original sequence accumulation method to reduce the interference of the irregular fluctuations of the original sequence and constructs the mixed kernel function based on the combination of RBF and Poly which takes both the learning and generalization ability of the model into account. The genetic algorithm is used to optimize the parameters of the LSSVM model. Through characteristic chromosome coding of the model parameters, the search space is established to obtain the optimal solution through fitness evaluation. The simulation results show that the model can effectively predict the network security situation with an accuracy of about 13% higher than that of HHGA-RBFNN and PSO-SVM.

Acknowledgements

This work was supported in part by Key Science and Technology Project of Jilin Province (20160204019GX) and National High-tech R&D (863) Program of China (2015AA015701).

References

1. T.Q. Huang, Y. Zhang, An approach to real time network security situation prediction, 2014, Journal of Chinese Computer Systems, vol. 35, no. 2, pp. 303-306.

2. C.X. Guo, Y. Su, A new optimized algorithm based on quantum evolutionary strategy for network security situation prediction, 2014, Journal of Chinese Computer Systems, vol. 35, no. 9, pp. 2083-2087.
3. K. Wu, Z.Y. Bai, Trusted network security situational awareness and forecast based on SPA, 2012, Journal of Harbin Institute of Technology, vol. 44, no. 3, pp. 112-118.
4. C.H. Tang, S.Z. Yu, Method of network security situation prediction based on likelihood BP, 2009, Computer Science, vol. 36, no. 19, pp. 97-100.
5. S.Y Wu, X.G Tian. Method for anomaly detection of user behaviors based on hidden Markov models, 2007, Journal on Communications, vol. 28, no. 4, pp. 38-43.
6. Y.L. Liu, D.G. Feng, Y.F. Lian, Network situation prediction method based on spatial-time dimension analysis, 2014, Journal of Computer Research and Development, vol. 51, no. 8, pp. 1681-1694.
7. S.S. Keerthi, C.J. Lin, Asymptotic behaviors of support vector machines with Gaussian Kernel, 2003, Neural Computation, vol. 15, no. 7, pp. 1667-1689.
8. S.H. Xiong, C.Y. Zhou, LSSVM prediction model for chaotic time series based on reduction strategy, 2011, Acta Scientiarum Naturalium Universitatis Sunyatseni, vol. 50, no. 1, pp. 53-57.
9. H.J. XIAO, X.L. Sang, Network security situation prediction based on hyper parameter optimization of relevance vector machine, 2015, Journal of Computer Applications, vol. 35, no. 7, pp. 1888-1891.
10. M.Y. Zhao, Y. Tang, C.Z. Fu, T. Ming, Feature selection and parameter optimization for SVM based on genetic algorithm with feature chromosomes, 2010, Control and Decision, vol. 25, no. 8, pp. 1133-1138.
11. Z.C. Wen, Z.G. Chen, Network security situation prediction method based on hidden Markov model, 2015, Journal of Central South University (Science and Technology), vol. 46, no. 10, pp. 3689-3695.
12. R.R. Xi, X.C. Yun, Y.Z. Zhang. An improved quantitative evaluation method for network security, 2015, Chinese Journal of Computers, vol. 38, no. 4, pp. 749-758.
13. N. Görnitz, M. Kloft, K. Rieck, Toward supervised anomaly detection, 2013, Journal of Artificial Intelligence Research, vol. 46, no. 2, pp. 235-262.
14. V. Dutt, Y.S. AHN, C. Gonzalez, Cyber situation awareness modeling detection of cyber attacks with in-stance-based learning theory, 2013, Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 55, no. 3, pp. 605-618.

15. R.Z. Fan, M.K. Zhou, Network security awareness and tracking method by GT, 2013, Journal of Computational Information Systems, vol. 9, no. 3, pp. 1043-1050.